

# WISHI Semantic Interop IETF 101 Hackathon

Planning and Materials

# WISHI Interop Planning

- Focus Areas
- Implementation Guidance
- Survey of standards, tools, and implementations
- Participant Questionnaire

# Focus Area: Semantic Interoperability and Hypermedia

- What to do => How to Do It
- Across the application workflow
  - Discovery
  - Configuration
  - Operation
- Description and classification model
  - Thing type
  - Capability
  - Interaction
  - Data Shape

# Semantic Description and Classification Model

- Layered descriptive model
  - Thing type (A Thing is a type of thing)
    - luminaire, washing machine, door
  - Capability (A Thing has a Capability)
    - on/off, brightness, opening, closing, sensing
  - Interaction (A Capability provides an Interaction)
    - open, turn on, read temperature
  - Data Shape (An Interaction exchanges a Data representation of a particular Shape)
    - Integer,  $0 \leq \text{value} \leq 255$

# Implementation Guidance

- Goals and scenarios related to use cases and application workflow
  - Discovery – First Objective
  - Configuration
  - Operation
  - Security, Accessibility, Privacy Considerations
- Survey of standards, tools, and implementations
  - Standards and community organizations
  - Representation Formats
  - Examples to be submitted and collected
  - Development tools
  - Open source implementations
  - Other interops, e.g. W3C plugfests

# Application Workflow

- Discovery
  - How does an application find what it needs to configure an instance of some functional capability, e.g. a thermostat in a room?
  - Inputs, outputs, controls, and settings
- Configuration
  - How do I select from multiple possibilities, how do I integrate diverse things into the application? e.g. device selection, units and scale adaptation
- Operation
  - How is information communicated and processed in the application? e.g. protocol adaptation, CoAP Observe, MQTT Subscribe

# Standards, Semantic Descriptions, and Serialization Formats

- OCF Resource formats, oneiota definitions
- LWM2M/IPSO Smart Object format
- WoT Thing Description
- iotschema Definitions; schemas and examples
- QUDT, SSN, SOSA
- CORAL
- SenML, CoRE Link-Format, HSML
- JSON-Hyperschema

# Example Implementations

- Thing Directory from Thingweb
- Node-Wot WoT Servient from Thingweb
- ARM mbed, LWM2M implementation for constrained devices, CoAP libraries
- OCF - iotivity CoAP Client and Server, C and nodejs
- Experimental
  - iot-toolkit (RDF+REST cross-proxy library in python, http, MQTT, CoAP)
  - MachineHypermediaToolkit (link-format+SenML, python)



# Semantic Repositories

- iotschema definitions
- Project Haystack

# Interop Participant Questionnaire

- Collect information from participants and implementers
- Enable exchange of ideas and concepts to converge on some common patterns
- Mostly optional but more information is better
  - Scenario description, what is the high level design pattern
  - Implementation description
  - What system roles are implemented; application, thing, proxy, directory, ?
  - How are discovery, configuration, and operation handled?
  - What protocols and representation formats are supported?
  - What semantic capabilities are needed for the application?
  - What semantic capabilities are provided by the things?

# Thing Directory

- <https://github.com/thingweb/thingweb-directory>
- REST API using OpenAPI (Swagger) definition
- POST to upload and register a Thing Description
- SPARQL endpoint using GET + query parameters to discover a Thing Description that matches a semantic specification
- Also the node-wot servient:
  - <https://github.com/thingweb/node-wot>

# Annotation Example: Thing Type, Capability, Interaction, Data Shape

```
"base": "coap://example.net:5683/",
"@type": [ "Thing", "iot:SensorThing", "iot:TemperatureCapability" ],
"name": "Temperature Sensor",
"interaction": [
  {
    "name": "Temperature",
    "@type": [ "Property", "iot:TemperatureProperty" ],
    "outputData": {
      "type": "object",
      "field": [
        {
          "name": "temperature",
          "@type": [ "iot:TemperatureData" ],
          "type": "number",
          "minimum": -50,
          "maximum": 100,
          "unit": "Celsius"
        }
      ]
    }
  }
]
```

# Example iotschema Definition

```
"type": "TemperatureCapability",
"id": "iotschema:TemperatureCapability",
"subClassOf": "iot:Capability",
"description": "Temperature Sensing and Control Capability",
"providesInteractionPattern": {
  "type": "Property",
  "subClassOf": "iot:Property",
  "name" : "TemperatureProperty",
  "@id": "iot:TemperatureProperty",
  "providesOutputData": {
    "type": "schema:Number",
    "@id": "iot:TemperatureData",
    "schema:valueName": "temperatureData"
    "schema:unitCode": { "@id": "iot:TemperatureUnit" },
    "schema:minValue": "schema:Number",
    "schema:maxValue": "schema:NUmber"
  }
}
```